1.prompt the user to enter a text or provide a file to count the words.

```
public class WordCount {
    public static void main(String[] args) {
        String inputString = "India Is My Country";
        int wordCount = inputString.split("\\s+").length;
        System.out.println("Number of words: " + wordCount);
    }
}
```

2.read the input text or file and store it in a string.

```
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Path;

public class ReadFileToStringExample {
    public static void main(String[] args) throws IOException {
        Path filePath = Path.of("C:\\Users\\HP\\Desktop\\gfg.txt");
        String fileContent = Files.readString(filePath);
        System.out.println(fileContent);
    }
}
```

3.split the string into an array of words using space or punctuation as delimiter.

```
String inputString = "Hello, world! This is a sample string.";
String[] words = inputString.split("[,.!?\\s]+"); // Split by spaces, commas, periods,
exclamation marks, or question marks
```

4.initialize a counter variable to keep track of the number of words.

```
public class WordCounter {
    public static void main(String[] args) {
        String sentence = "This is a sample sentence with several words.";
        String[] wordsArray = sentence.split("\\s+");
        int wordCount = wordsArray.length;
        System.out.println("Number of words: " + wordCount);
    }
}
```

5.iterate through the array of words and increment the counter for each word encountered.

```
import java.util.*;

public class WordCounter {
    public static void main(String[] args) {
        // Example input: an ArrayList of words
        List<String> wordList = new ArrayList<>();
        wordList.add("apple");
        wordList.add("banana");
        wordList.add("apple");
        wordList.add("cherry");
        wordList.add("banana");

        // Create a map to store word occurrences
        Map<String, Integer> wordCount = new HashMap<>();

        // Iterate through the wordList
        for (String word : wordList) {
            // Get the current count (or initialize to 0 if not found)
            Integer count = wordCount.get(word);
            // Increment the count by 1
            wordCount.put(word, (count == null) ? 1 : count + 1);
        }

        // Print word occurrences
        for (Map.Entry<String, Integer> entry : wordCount.entrySet()) {
            System.out.println(entry.getKey() + ": " + entry.getValue());
        }
    }
}
```

6.display the total count of words to the user.

```
import java.util.Scanner;

public class WordCounter {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        printWordsAndLines(input);
    }

    public static void printWordsAndLines(Scanner scanner) {
        int words = 0;
        int lines = 0;

        while (scanner.hasNextLine()) {
            String line = scanner.nextLine();
            String[] wordsInLine = line.split("\\s+");
            words += wordsInLine.length;
            lines++;
        }

        System.out.println("Total words: " + words);
        System.out.println("Total lines: " + lines);
    }
}
```

7.ignoring common words or stop words.

```
@Test
public void whenRemoveStopwordsManually_thenSuccess() {
    String original = "The quick brown fox jumps over the lazy dog";
    String target = "quick brown fox jumps lazy dog";
    String[] allWords = original.toLowerCase().split(" ");
    StringBuilder builder = new StringBuilder();
    for (String word : allWords) {
        if (!stopwords.contains(word)) {
            builder.append(word);
            builder.append(' ');
        }
    }
    String result = builder.toString().trim();
    assertEquals(result, target);
}
```

8.providing statistics like the number of unique words or the frequency of each word.

```
import java.util.HashMap;
import java.util.Map;

public class WordFrequencyCounter {
    public static void main(String[] args) {
        String input = "Java is great and Java is powerful";
        String[] words = input.split(" ");

        Map<String, Integer> wordFrequency = new HashMap<>();

        for (String word : words) {
            wordFrequency.put(word, wordFrequency.getOrDefault(word, 0) + 1);
        }

        // Print word frequencies
        for (Map.Entry<String, Integer> entry : wordFrequency.entrySet()) {
            System.out.println(entry.getKey() + ": " + entry.getValue());
        }
    }
}
```

9.implementing input validation to handle empty inputs or file errors.

```
import java.util.Scanner;
```

```
public class InputValidationExample {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a file name: ");
        String fileName = scanner.nextLine().trim(); // Remove leading/trailing spaces

        while (fileName.isEmpty()) {
            System.out.println("File name cannot be empty. Please try again.");
            System.out.print("Enter a file name: ");
            fileName = scanner.nextLine().trim();
        }

        // Now you have a non-empty file name
        System.out.println("You entered: " + fileName);
    }
}
```

10.adding a graphical user interface (GUI) for a more user-friendly experience.

```
import javax.swing.*;

public class FirstSwingExample {
    public static void main(String[] args) {
        JFrame frame = new JFrame("My First GUI"); // Create a JFrame
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(300, 300); // Set window size

        JButton button = new JButton("Press"); // Create a button
        button.setBounds(130, 100, 100, 40); // Set button position and size
        frame.add(button); // Add button to the frame

        frame.setLayout(null); // Use no layout manager
        frame.setVisible(true); // Make the frame visible
    }
}
```

….